IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.DOI

Data Provenance with Retention of Reference Relations

CHUNDONG WANG[1], LEI YANG[1], YIJIE WU[2], YUDUO WU[2], XIAOCHUN CHENG[3](SENIOR MEMBER, IEEE), ZHAOHUI LI*[2], ZHELI LIU[2]

[1]Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin 300384, China (e-mail: michael3769@163.com, guashushang89757@163.com)
[2]College of Cyberspace Security, Nankai University, Tianjin 300350, China (e-mail: 2120170537@mail.nankai.deu.cn,wuyuduo0508@163.com,lizhaohui@nankai.edu.cn,liuzheli@nankai.edu.cn)
[3]Department of Computer Science, Middlesex University, London NW4 4BT, U.K

Corresponding author: Zhaohui Li (e-mail:lizhaohui@nankai.edu.cn).

This work was supported in part by the Foundation of the Educational Commission of Tianjin, China, under Grant 20130801, in part by the General Project of Tianjin Municipal Science and Technology Commission under Grant 15JCYBJC15600, National Natural Science Foundation of China (No. 61672300), National Natural Science Foundation of Tianjin (No. 16JCYBJC15500) and Ministry of Education and China Mobile Scientific Research Foundation (NO. MCM20170403).

ABSTRACT

With the development of data transactions, data security issues have become increasingly important. For example, the copyright authentication and provenance of data have become the primary requirements for data security defence mechanisms. For this purpose, this paper proposes a data provenance system with retention of reference relations (called RRDP), which can enhance the security of data service in the process of publishing and transmission. The system model for data provenance with retention of reference relations adds virtual primary keys using reference relations between data tables. Traditional provenance algorithms have limitations on data types. This model has no such limitations. Added primary key is auto-incrementing integer number. Multi-level encryption is performed on the data watermarking to ensure the secure distribution of data. The experimental results show that the data provenance system with retention of reference relations has good accuracy and robustness of the provenance about common database attacks.

INDEX TERMS Data provenance; reference relations; primary foreign key; database watermarking

I. INTRODUCTION

NOWADAYS, the development of social media information and networking has led to explosive growth of data service. Data transactions [1] are growing with the generation, exchange, management and application of data streams. As the volume of data transactions increases, the security issues become prominent. In this case, Trusted Computing (TC) is a good solution. Trusted Computing is the concept of adding trust to the information society. This technology can enhance the overall security, privacy and trustworthiness of various computing devices, build trusting society, and improve the level of information security. There are also many other existing protection measures under the data sharing environment [2]. A directly revocable attribute-based encryption scheme (ABE) [3] and a revocable IBE (identity-based encryption) scheme [4] are proposed to protect the secure storage of data in cloud environments. Fan [5] has studied the effect of transmit power allocation among jammer on security performance. Cao [6] proposes a novel coverless information hiding method based on MSIM(molecular structure images of material) to guarantee the security of information transmission. Encrypted transmission based security scheme in IoT is studied by Jhaveri [7] and Wang [8]. The operational scheme of functional Android malware detection [9], [10] is also getting more and more attention.

Data service can be vulnerable to malicious dissemination in the process of circulation. Once the transaction data is illegally stolen, leaked, or tampered, it may pose a serious threat to the security of involved individuals or enterprises [11]–[13]. With the rapid development of emerging technologies, such as the Internet of Things (IoT), artificial intelligence, cloud and fog computing, new challenges have raised for the secure transmission of data. Traditional TC technology is not enough to meet current security needs. People's online behavior or digital footprints can be recorded by the data transaction platform, such as data consumption habits or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

friends networking information, etc. Once the data is leaked or stolen in the process of the circulation, not only will it cause a certain degree of damage to the user, but will also infringe the copyright of the data in the trading platform [14]. Therefore, how to trace the source and locate the leaker quickly after data loss is important to ensure the security of data service [15]. Data provenance technology is an effective measure to solve this problem.

A. DATA PROVENANCE

Data provenance is a technique with traceability, which can reproduce the historical state and evolution of data based on tracing paths. Previous research on provenance has focused only on database systems [16], [17], file systems [18], workflow systems [19]–[22] , and operating systems [23]. This paper focuses on the data provenance technology in database system.

Normal methods. Early provenance techniques mainly include annotation method [24] and query inversion method [25], the annotation method could record the relevant information to achieve the historical state of the data. Since the annotation method is not suitable for fine-grained data, especially the data provenance in big data. Fan [25] proposes a reverse query method, which inverts the query by reverse query or constructing a reverse function . This method should be calculated when needed, so it is called the lazzy method. Alawini [26] mainly shows the connection between data citation and data provenance.

Typical methods for database. Database fingerprint technology can also effectively solve the problem of provenance. Most database fingerprinting techniques are designed based on database watermarking techniques. Li et al. [27] proposes a fingerprint relation database extraction technique. In [28], a new fingerprint structure NoFiA is proposed to protect the relational data in the database system from illegal copying and redistribution. In [29], an availability constraint is proposed for the fingerprint relational database. The declarative language defines an availability-constrained digital watermark and fingerprint relational database. They optimize watermark embedding through search. Liang [30] proposes a decentralized and trusted cloud data provenance architecture by blockchain. Fernanda presents the progger (provenance logger), a kernel-space logger which potentially empowered a cloud stakeholders to trace their data.

After referring to the relevant literatures of various existing data provenance techniques, we find that there are still many problems in the existing technical solutions.

- **Large storage requirement.** Although the early data provenance methods [31]–[33] are easy to implement and manage, it is only suitable for small systems. However, it is difficult to provide detailed traceability information for the large systems. Additional storage space is required to store the tag information. These methods are less efficient and have some limitations.
- **Rely on data type.** The data provenance methods using database watermarking technology [34]–[37] have great limitations on data types. These current methods mainly focus on numerical data. Gupta's DEW technique [38] and Farfoura's PE-1 technique [39] introduces the differential extension, which could achieve the reversibility of the watermark embedding process. This method could ensure the availability of data to the utmost extent. However, it embeds watermark information by changing its redundant bits. The data types are more diverse in practical applications, and it is difficult to find effective redundant space without destroying the integrity of data and its reference relations.
- **Weak robustness.** The existing data provenance techniques [40] are suitable for specific application scenarios. Jawad's GADEW technique [41] is based on the idea of difference expansion and utilizes genetic algorithm (GA) to improve watermark capacity and to reduce distortion. But its robustness under the alteration attacks and deletion attacks is not very good. There is no single solution can defend against most attacks. In practice, a combination of multiple watermarking algorithms is generally used to resist more attacks.

Each technique has its own benefits, drawbacks, issues and limitations. There is no better way to solve all of the above problems effectively.

B. MOTIVATION

This paper found two important rules: (1) Tables are associated by primary foreign keys, which are usually not valid data, but auxiliary data. (2) In the previous database relations (tuple), the value of one column is closely related to the value of other columns. In general, some transformations will destroy their relevance. As a result, many algorithms cannot be applied.

Based on the above rules, we change the primary foreign key without changing the data reference relations between the tables, then we can apply a certain implicit rule as the embedded watermark information on the generated primary foreign key to preserve the original state of data. Once the data is leaked, the source of the data leak can be traced by the extracted watermark. This solution can be implemented both on static datasets and dynamic datasets.

There are three goals: 1)Find effective redundant space in the database; 2) Embed watermark information without affecting the use of data; 3) Trace the source efficiently after data leakage. The study found that the primary foreign key has many advantages, it can be used to meet all the basic requirements mentioned above: 1) Position of embedding watermark is more flexible, we can dynamically add the primary foreign key value without too much extra storage space, this method effectively solves the problem of large amount of data storage in current data provenance technology. It is not only suitable for small systems, but also conforms to the background of today's big data; 2) The scheme uses table structure transformation to imply some certain rules as the watermark information, which could break through the limitations of data types in traditional provenance techniques; 3)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

TABLE 1: Comparison between RRDP and previous scheme

Scheme	Elapsed time overheads		Robustness			Data type	
	Embed	Detect	Alteration attack	Insertion attack	Deletion attack	Numeric	Non-numeric
Previous work							
Gupta [38]	$O(n)$	$O(n)$	medium	medium	higher	✓	×
Jawad [41]	$O(n)$	$O(n)$	low	low	medium	✓	×
Farfoura [39]	$O(n)$	$O(n)$	high	high	higher	✓	×
This work							
RRDP	$O(n)$	$O(\log n)$	higher	higher	higher	✓	✓

Due to the association of the primary foreign key, the attacker generally doesn't destroy the primary foreign key between the tables, so it can resist various database attack types, such as alteration attacks, insertion attacks, and deletion attacks. Compared to the previous schemes, this scheme has better robustness.

C. OUR CONTRIBUTION

We compare the RRDP with several typical database watermark schemes in terms of time overhead, robustness and data type. Table 1 summarizes the comparison of the RRDP scheme with the previous schemes.

In the comparison of the time consumption of the above schemes, we don't consider the distribution of multi-level and multi-objects, we mainly analyzes the complexity of data embedding watermark and detecting watermark. The time complexity of embedding watermark in our scheme is O(n). When performing watermark detection, the RRDP scheme doesn't need to verify all the tuples, so we think that the computational complexity is about O(log n), which is smaller than the minimum value of all the above algorithms O(n); In terms of the robustness of the algorithm, we mainly analyze the resilience of alteration attacks, insertion attacks and deletion attacks. It is found that only the RRDP scheme can effectively resist these three types of attacks; For data types, the previous watermarking algorithms mostly rely on data types, but the RRDP algorithm is suitable for various data types.

In this paper, a data provenance algorithm with retention of reference relations is proposed. The main contributions of this paper are as follows:

- Our proposed algorithm doesn't require a lot of extra space when embedding and storing traceability information. The algorithm is not only suitable for small systems, but also for today's big data background.
- The generation rules of the primary foreign key is used to embed the watermark information without changing the original data. Therefore, we don't need to consider the specific data type, which breaks the limitation of the general database watermarking algorithm on data type processing.
- Our data provenance scheme can resist most common database attack types, such as alteration attacks, insertion attacks and deletion attacks, so it has good robustness.

II. PRELIMINARIES

A. NOTATIONS

Throughout this paper, the notation D_O,D_P,D_U denote the original data owner, watermark processor and data user respectively. Other parameters include:
-A is the original database.
-The set $\{a_1^m, a_2^m, ..., a_i^m\}$, $\{b_1^m, b_2^m, ..., b_j^m\}$, $\{c_1^m, c_2^m, ..., c_k^m\}$ denote the different split-tables which are distributed to the different m-level data users.
-i, j and k denote the number of the split-tables respectively ($i \neq j \neq k$).
-The set $\{x_1', x_2', ..., x_t'\}$ denotes the leaked tables.
-K_i denotes the key corresponding to different data users.
-The set $\{T_1, T_2, ..., T_i\}$ denotes the 0-1 attribute tables.
-T' denotes the leaked 0-1 attribute table.
-P denotes the original auto-incremented integer column.
-P_e denotes the encrypted auto-incremented integer column.
-P_es denotes the auto-incremented integer column after sorting P_e.

Definition 1. (0-1 attribute table). This 0-1 attribute table describes the attributes owned by each of the split tables. The header row is the name of all the attributes in the original table $attribute1, attribute2,$ the header column is the name of each table after split $\{a_1, a_2, ...\}$. If the split table a_1 has $attribute2$, the data in the corresponding table is 1, otherwise 0.

Definition 2. (K-Tree). The hierarchy relations of data distribution described by this tree structure. The data of the node in the tree stores the code of each data user $\{U_1, U_2, ...\}$, where the root node represents the owner of the original data, and the parent of each child node represents the data distributor of the previous level.

Definition 3. (U-Key table). The table describes the data user's name, code name, the corresponding key and 0-1 attribute table.

Definition 4. (3DES). 3DES is a mode of the DES encryption algorithm which uses three 56-bit keys to encrypt data three times.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

B. SYSTEM MODEL

The data of the data owner D_O will be collected, transmitted, processed and stored before being distributed. Data processing unit D_P is the core of the model, whose function is to watermark the data on the premise of storing data users' information and distribution, and to trace the source of the data after the data leaks. There are also some other entities in this system, such as data users D_U, including the company and its subsidiaries.

Figure.1 shows an example of data distribution, theft and traceability. In the Figure.1, D_O gives the original data A to D_P. There are three data users distributed at level 1.

D_P performs three different types of splits and changes on the table structure of A, then we get the processed table set $\{a_1^0, a_2^0, ..., a_i^0\}, \{b_1^0, b_2^0, ..., b_j^0\}, \{c_1^0, c_2^0, ..., c_k^0\}$. Each company's corresponding key K_i is used to encrypt it, and get the table set $\{a_1^1, a_2^1, ..., a_i^1\}, \{b_1^1, b_2^1, ..., b_j^1\}, \{c_1^1, c_2^1, ..., c_k^1\}$, which are distributed separately to data User1, User2 and User3.

Once D_O discovers the data leak, the leaked table set $\{x_1', x_2', ..., x_t'\}$ is handed over to D_P for traceability. D_P compares the 0-1 attribute table T of the leaked table set with its stored T_i. Then the code of the data user is determined based on the U-Key table, and the circulation of its distribution data is determined through the position of the code in the K-Tree. Finally, the traitor is identified by using the key in the U-Key table to compare the encrypted data and the leaked data.

C. RRDP

The following algorithm can be used to describe a data provenance scheme, which is retained based on the reference relations.

- **RRDP. Setup**(A): Given a original database A, the output will be a set of split tables $\{a_1^0, a_2^0, ..., a_i^0\}$.
- **RRDP. WMEmb**$([a], K_i)$: Given a set of split tables $[x']$ and encryption key K_i , the output will be a set of new tables $[a^1]$.
- **RRDP. TRPro**$([x], [a^0], K_i)$: Given the leaked tables $[x']$, the original tables $[a^0]$ and the encryption key K_i, the outputs will be the encrypted table $[x]$. Compare $[x]$ with $[x']$ and trace the traitor.

Security model. In this model, some users are considered curious and malicious. They may disclose all or part of the records, but they will not maliciously replace or delete all the primary foreign keys. In general, this model can resist the common database attacks, such as tuple attacks and attribute attacks.

III. OUR SCHEME

We will introduce a new data provenance method which preserves the reference relations. First, the primary foreign key relations between the tables is changed by the self-incremented primary key, and a new order is generated using a key space-based replacement algorithm, which is the watermark information. Once the data is leaked, the leaker can be verified by continuously encrypting it from the source. We named it $RRDP$. This algorithm is aimed at the related data in the database, it includes three parts: primary foreign key alter, watermark embedding, tracking and provenance.

A. PRIMARY FOREIGN KEYS ALTER

The primary foreign key change algorithm is described in Figure 2. D_P splits and alters the original data table structure of D_O, and adds or replaces its primary foreign key to each of the split tables, thus forming a new table structure which retains the reference relations. Different users' data will be split into different types of table structures to facilitate efficient provenance after data leakage.

Input: D_O inputs a set of original database tables A of N tuples.

Output: D_P outputs a set of altered-tables $[a^0]$.

1. Splitting tables: According to the actual demands, the tables are divided into two types: single-table and multi-table. At the same time, for the different distribution units, the corresponding table can be split into different structures and number T so that the range can be quickly located when traced.

a) Single table splitting: The vertical splitting method is adopted when dealing with a single table. That is, table A is divided into different database tables $[a^0]$ according to its modules and functions without destroying the third normal form.

b) Multi-table splitting: When there is no primary foreign key association or weak correlation between multiple tables, we use the single table split method in (a) to split multiple tables.

2. Primary-foreign key alter: The structure of the database table $[a^0]$ is changed. Add a sequentially increasing integer column P as the self-incrementing primary key while retaining the primary key of the table a_1^0. Then the foreign key of the table $[a^0]$ is replaced by the integer primary key added in the table a_1^0, in order to retain the primary foreign key relations between the tables.

3. Storing the 0-1 attribute table: Store the situation of attribute existence by each split table as a 0-1 table. The header row is the name of all the attributes in the original table $\{atteibute1, attribute2, ...\}$. The header column is the name of each table after the split $\{a_1, a_2, ...\}$. If the split table a_1 has $attribute2$, the data in the corresponding table is 1, otherwise 0.

$A = \{Name, passeword, telephone, email, address\}$ is a table in the original database. It is split into $a_1 = \{Name, password, email\}$ and $a_2 = \{Name, telephone, address\}$ as the above-mentioned steps; Then change the structure of the two tables after the split, the sequentially incremented ID integer column $P = \{1, 2, 3, 4, 5\}$ is added as the self-incrementing primary key while retaining the primary key $Name$ of the table a_1. Next, replace the foreign key Name in table a_2 with the ID integer column $P = \{1, 2, 3, 4, 5\}$ which is added in table a_1. Table 2 is the corresponding 0-1 attribute table we stored.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

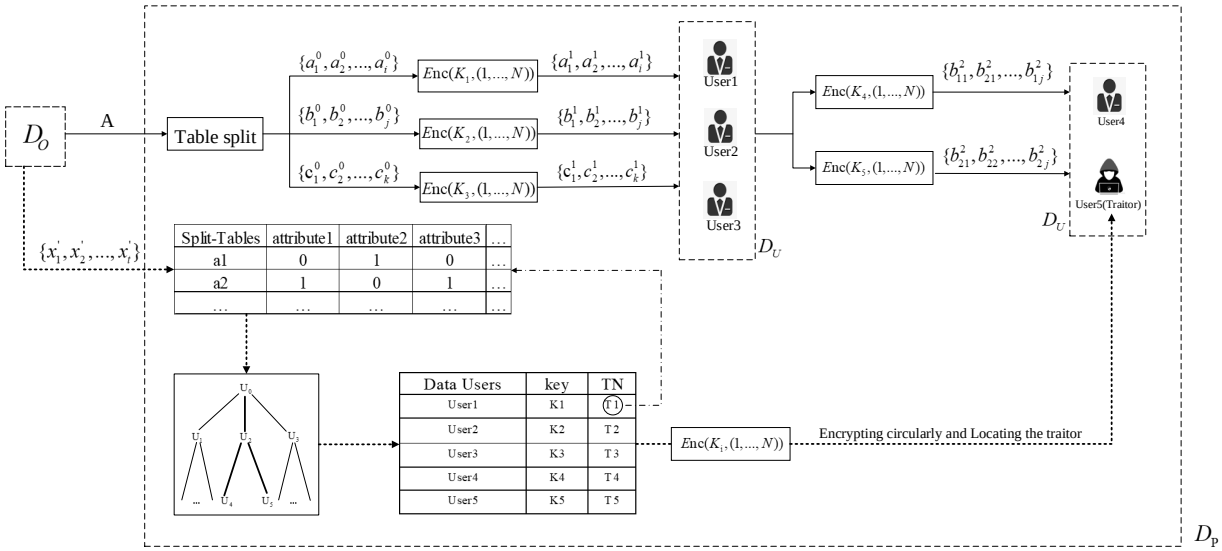

FIGURE 1: Example of data distribution, theft and traceability

TABLE 2: a_1, a_2 0-1 attribute table

T_1	Name	Password	Telephone	Email	Address
a_1	1	1	0	1	0
a_2	0	0	1	0	1

TABLE 3: U-Key table

User name	User code	Key	TN
User1	$U_1(1)$	$K_1(A)$	T_1
User2	$U_2(2)$	$K_2(B)$	T_2
User3	$U_3(3)$	$K_3(C)$	T_3
User4	$U_4(4)$	$K_4(D)$	T_4
User5	$U_5(5)$	$K_5(E)$	T_5

B. WATERMARK EMBEDDING

This watermark embedding algorithm is described in Figure 1 . It is divided into two steps: key generation and storage and key space-based replacement.

Input: D_P inputs a set of altered-tables $[a^0]$ of N tuples, encryption key . K_i

Output: D_P outputs a set of watermarked-tables $[a^1]$.

1. Key generation and storage: Map the string of D_U names as a key K_i before the data publishing. The number of unit U_i and the key K_i are stored in the U-Key table one-to-one, and U_i is stored in the K-Tree according to the hierarchical relations of the distribution.

2. Key space-based replacement: The key space-based permutation algorithm $(k, 0...N)$ is used to rearranged the tuples and distribute the rearranged data to the affiliate.

a) According to the key K_i, the basic block cipher E (eg, $3DES, AES$, etc.) is used to encrypt the self-incrementing integer column $P = \{p_1, p_2, ..., p_m\}$. Which calculates the tuple $I = (E_k(p_1), E_k(p_2), ..., E_k(p_m))$;

b) Each component $E_k(i) \in I$ is a different binary string whose length depend on the packet size, and it is sorted according to the numerical relations. Then obtain the sort value r_i corresponding to $E_k(i)$, replace the component $E_k(i)$ with its corresponding sort value and get the integer column $P_e = (p_1{}', p_2{}', ..., p_m{}')$;

c) The tuples in table are rearranged in the ascending order of the encrypted integer column $P_e = (p_1{}', p_2{}', ..., p_m{}')$ to obtain a rearranged table.

Select $3DES$ as the basic block cipher E, the key is $k_1(A)$. In order to facilitate subsequent cryptographic operations, we take the simplest self-incrementing integer column of number 5. Then we encrypt the ID integer column $\{1, 2, 3, 4, 5\}$ according to the key $k_1(A)$, calculate $E_{k_1}(1) = 5753214040127886486$; $E_{k_1}(2) = 3011020924204599246$; $E_{k_1}(3) = 0450904572176669844$; $E_{k_1}(4) = 7453689656944979882$; $E_{k_1}(5) = 6219725333994694406$ to obtain the tuple $I = (E_k(p_1), E_k(p_2), ..., E_k(p_m))$. They are sorted according to the numerical relations, thus we get the encrypted sequence $P_e = \{3, 2, 1, 5, 4\}$. Reorder the tuples in the table by the ascending order of the encrypted ID integer column $P_e = \{3, 2, 1, 5, 4\}$, the mapping relations between the self-increase integer column P in the original table and the encrypted integer column P_e is the embedded watermark information. Further, D_P stores the unit number U_1 of User1 and the key $K_1(A)$ in the U-Key table, and it also stores U_1 in the K-Tree according to the hierarchy of its distribution (U_0 represents the data owner D_O). As shown in Figure 2, according to the same processing method as above, the data is distributed to User2, ..., User5, then the U-Key table(Table 3) and K-Tree (Figure. 2) is described as follows.

C. TRACKING AND PROVENANCE PHASE

Once the data is leaked, the original data is encrypted continuously by the key space-based replacement algorithm and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

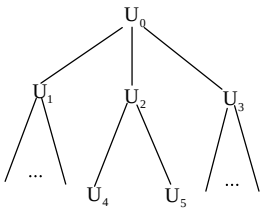

FIGURE 2: K-Tree

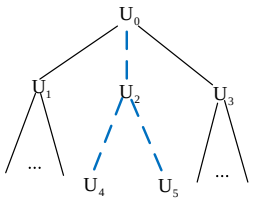

FIGURE 3: K-Tree (Leak path)

information stored by D_P, and then the encrypted sequence is compared with the sequence of the leaked data. Finally the leaker is found.

Input: D_P inputs a set of leaked-tables $[x']$, original data tables $[a^0]$, U-Key table, K-Tree, 0-1 attribute table.

Output: D_P outputs the code of traitor U_i.

1. Comparing the attributes: Extract the attributes' situation of the leakage table $[x']$, compare it with the 0-1 attribute table stored by D_P, then T_i is determined, the branch structure T_b where the data in the K-Tree is located is obtained by querying the U-Key table.

2. Encryption and tracking: The U-Key table is queried to obtain all T_i of T_b According to T_b. Then, the distribution path in T_b is simulated by the encryption algorithm in 3.2, and the original table $[a^0]$ is encrypted multiple times using the involved K_i.

3. Verification and positioning: Compare each encrypted table $[a^i]$ with the leaked table $[x']$. After the comparison is successful, determine the code of traitor U_i according to the U-Key table.

Assume D_O finds a leaked database table $[x']$ as $x_1 = \{Name, password\}$, $x_2 = \{email\}$ and $x_3 = \{telephone, address\}$. The corresponding 0-1 attribute table obtained by $[x']$ is shown in Table 3. Compare T_x with the 0-1 attribute table stored by D_P, then we get $x = 2$.

By querying the U-Key table, all K_i=$\{K_2(B), K_4(D), K_5(E)\}$ in the branch structure T_b where the data in the K-Tree is located (the bold line in Figure.3) is obtained. Then all the data leakage path is known by T_b: ① $User2$; ② $User2 \rightarrow User4$; ③ $User2 \rightarrow User5$. The key space-based replacement algorithm and K_i is used to perform encryption comparison according to the above three paths, and locate the traitor User4 at last.

D. ANALYSIS OF RRDP

In this section, the RRDP algorithm will be analyzed from three aspects: algorithm efficiency, accuracy and security.

1. Efficiency analysis.

In order to meet the needs of today's big data, provenance algorithms must ensure higher efficiency to reduce the time consumption and cost requirements. For the RRDP algorithm, we mainly consider its computational complexity for first-level distribution and multi-level distribution.

a) First-level distribution

For the first-level distribution, the encrypted data should be compared with the leaked data. At this time, the computational complexity of the provenance is O(n). However, in the actual provenance process, our scheme doesn't need to compare all the tuples, only 0.1% of the total number of tuples could ensure the accuracy of provenance. Therefore, the computational complexity of the provenance process of RRDP is smaller than O(n).

b) Multi-level distribution

For the multi-level distribution, in order to quickly determine the leaked branch, this scheme first introduced the 0-1 attribute table. We store the 0-1 attribute table of each user in the U-key table. The data distribution path is stored as K-Tree in the form of a multi-way tree. Once the data leaks, the 0-1 attribute of the leaked data is first extracted and compared to the 0-1 attribute table we stored. Then we can quickly locate the leaked branch in K-Tree. This algorithm makes the previous traversal of the entire tree pair into a traversal of a subtree, which greatly reduces the computational complexity of provenance.

2. Accuracy analysis

For provenance algorithms, accuracy is an important metric to measure performance. For the RRDP algorithm, it uses the primary foreign key association between the tables instead of the original data values to embed the watermark information, and it does not require additional storage space. Therefore, no matter how the amount of data grows, as long as the corresponding relationship between several rearranged primary foreign keys in the leakage table and the remaining tuples is verified at the phase of provenance, it can be determined whether it is the traitor. The common operation in the database does not affect the accuracy of the algorithm at all. The accuracy of this algorithm has always been at a high standard.

3. Security analysis.

The RRDP scheme uses the encryption of the primary foreign key to embed the watermark information. When an attacker maliciously deletes all primary keys, it will affect the correlation between data, thereby destroying the use of data. Therefore, in the above security model, we assume that the attacker doesn't maliciously delete or change all the primary foreign keys. Then, we analyze the ability of the algorithm to defend against three common database attacks as follows.

a) Alteration attacks and insertion attacks

Generally, alteration attacks and insertion attacks will not change the mapping relations between the primary foreign key and the other attributes. Therefore, inserting or changing any tuples does not affect the embedded watermark information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

b) Deletion attacks

Although the deletion attacks delete part of the data and the primary foreign key, the reference relations between the primary foreign key and the attribute of the remaining data can still be traced. Only 0.1% of the tuples could trace the traitor. Therefore, the algorithm can also effectively defend against deletion attacks.

By analyzing the resilience of the algorithm under the three common database attacks, this RRDP scheme presents better robustness.

IV. EXPERIMENTAL EVALUATION

Considering that the algorithm is mainly applicable to data provenance in the database. In this section, the encryption time and database processing time is verified. In general, the database often suffer from some attacks. In order to ensure that the algorithm can still trace the source accurately under database attacks, the robustness of the watermark information under the external attacks also needs to be verified. In this experiment, we mainly consider alteration attacks, insertion attacks and deletion attacks.

A. EXPERIMENT SETUP

In this section, performance evaluation is provided for the proposed data provenance algorithm. To test the validity and robustness of this algorithm, we perform experiments on a computer running Windows 10 Professional with 3.6 GHz CPU and 8GB RAM. We apply our algorithms to the real dataset, available from https://www.kaggle.com/donorschoose/io/data, This dataset contains 72,994 tuples and 9 attributes. We select 10,000 tuples of them. Algorithms are implemented on Visual Studio Platform version 2015 using mysql's API for C++ to visit MYSQL database.

B. DATA PUBLISHING EFFICIENCY

We measure the time of data encryption and the time of database processing. In the experiment shown in Figure.4, We tested its performance using tuples of size 1000, 10,000 and 100,000. For different size of the tuples, we tested three times separately and counted its data encryption time and database processing time.

Figure 4 and 5 show that when the number of tuples is 1000, the histograms of the three colors represent the three measurements under the same parameters. The average value of the data encryption time (it refers to the time required for the self-incrementing primary key to be encrypted by FPE) is 38.67 ms, and the average value of database processing time is 6681ms. As the number of tuples increases, the encryption time and database processing time increase linearly.

C. ACCURACY OF PROVENANCE

In order to balance efficiency and accuracy, we set the similarity threshold γ = 0.1%*N (number of tuples). For example, when the number of tuples is 10,000, we believe that the renegade can be determined when the leakage table

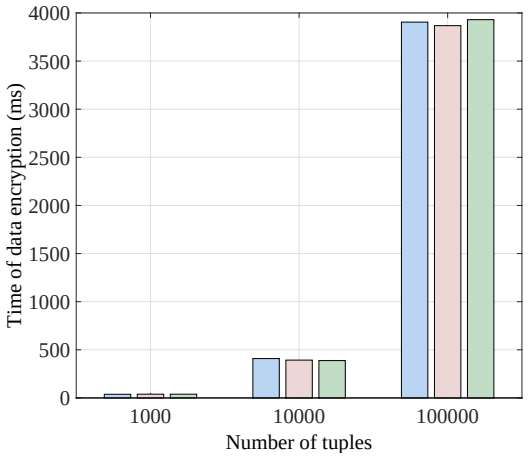

FIGURE 4: Data encryption time

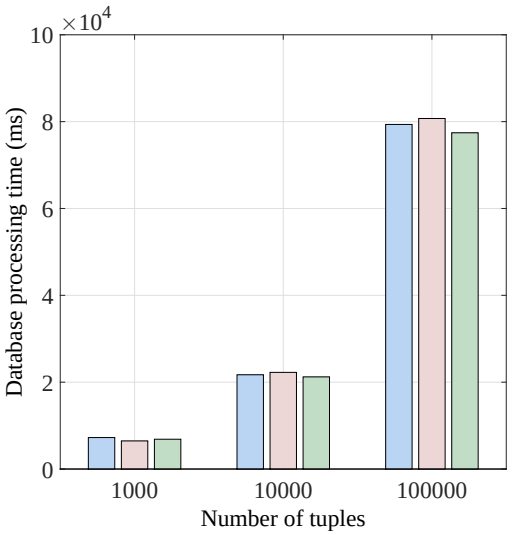

FIGURE 5: Database processing time

is identical to the original processed table with at least 10 data. Table 4 shows the provenance accuracy under different number of tuples. It can be seen that when the number of tuples is 1000, 10000, 100000, the provenance accuracy reaches 100%. When the number of test tuples is 1000000, the provenance accuracy rate is 98.7%.

TABLE 4: Provenance accuracy rates with different data sizes

Number of tuples	Provenance accuracy rate(%)
1000	100
10000	100
100000	100
1000000	98.7

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

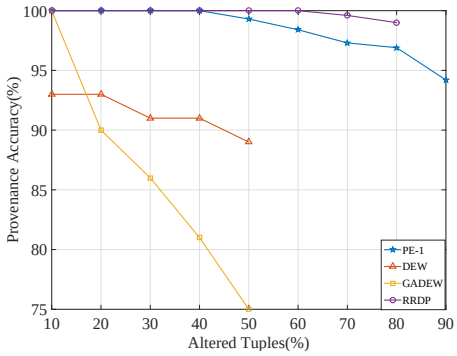

FIGURE 6: Accuracy of provenance with alteration attacks with different techniques

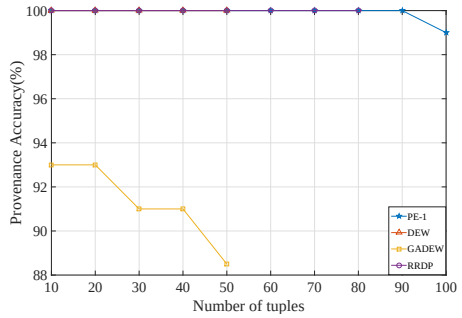

FIGURE 7: Accuracy of provenance with insertion attacks with different techniques

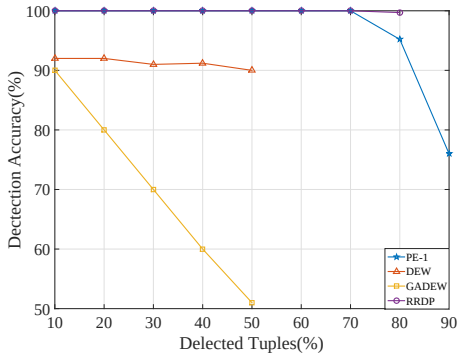

FIGURE 8: Accuracy of provenance with deletion attacks with different techniques

D. ROBUSTNESS

Accuracy of provenance is analyzed with alteration, insertion, and deletion attacks for DEW technique [38], GADEW technique [41], PE-1 technique [39] and our RRDP technique. The results are shown in Figure 6-8.

From figure 6-8, we can see that our RRDP scheme shows better robustness under these three common database attacks. It can be seen from the figure, the provenance accuracy of RRDP is basically maintained above 97% under the alteration attacks, when the percentage of the alteration tuples exceeds 60%, the provenance accuracy begins to decrease. Under the insertion attacks and the deletion attacks, the RRDP's provenance accuracy has been maintained at 100%.

V. CONCLUSION

In this paper, a data provenance method with retention of reference relations is introduced, which is used to track the leaker and to effectively analyze the responsibilities after data leakage. Compared with many previous provenance algorithms, this method doesn't directly modify the original data when embedding the watermark information. Instead, it uses the primary foreign key association between the tables in the database, and it adds a virtual primary foreign key to maintain the reference relations between the tables. The virtual primary key is a list of auto-incrementing integer. For different users, the system encrypts their virtual primary keys with different keys. In the provenance phase, the system constantly encrypts the virtual primary key of the original table structure and compare the data with the leaked data, then the leaker can be determined and the data leakage path can be obtained. This system can accurately track leakers without destroying the usefulness of the data. The system is a defense against relevant database attacks because of its good robustness.

REFERENCES

[1] M. Kamran and M. Farooq, "A comprehensive survey of watermarking relational databases research," arXiv preprint arXiv:1801.08271, 2018.
[2] Z. Liu, T. Li, P. Li, C. Jia, and J. Li, "Verifiable searchable encryption with aggregate keys for data sharing system," Future Generation Computer Systems, vol. 78, 2017.
[3] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," Cluster Computing, vol. 20, no. 3, pp. 2385–2392, 2017.
[4] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," Ieee Transactions on computers, vol. 64, no. 2, pp. 425–437, 2015.
[5] L. Fan, X. Lei, N. Yang, T. Q. Duong, and G. K. Karagiannidis, "Secure multiple amplify-and-forward relaying with cochannel interference," IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 8, pp. 1494–1505, 2016.
[6] Y. Cao, Z. Zhou, X. Sun, and C. Gao, "Coverless information hiding based on the molecular structure images of material," Computers, Materials & Continua, vol. 54, no. 2, pp. 197–207, 2018.
[7] R. H. Jhaveri, N. M. Patel, Y. Zhong, and A. K. Sangaiah, "Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial iot," IEEE Access, vol. 6, pp. 20 085–20 103, 2018.
[8] C. Wang, J. Shen, Q. Liu, Y. Ren, and T. Li, "A novel security scheme based on instant encrypted transmission for internet of things," Security and Communication Networks, vol. 2018, 2018.
[9] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine learning based android malware detection," IEEE Transactions on Industrial Informatics, 2018.
[10] K. Riad and L. Ke, "Roughdroid: Operative scheme for functional android malware detection."
[11] H. Guo, Y. Li, A. Liu, and S. Jajodia, "A fragile watermarking scheme for detecting malicious modifications of database relations," Information Sciences, vol. 176, no. 10, pp. 1350–1378, 2006.
[12] M. Kamran and M. Farooq, "A formal usability constraints model for watermarking of outsourced datasets," IEEE transactions on information forensics and security, vol. 8, no. 6, pp. 1061–1072, 2013.
[13] S. Rani, D. K. Koshley, and R. Halder, "A watermarking framework for outsourced and distributed relational databases," in International Conference on Future Data and Security Engineering. Springer, 2016, pp. 175–188.

[14] M. E. Farfoura, S.-J. Horng, J.-L. Lai, R.-S. Run, R.-J. Chen, and M. K. Khan, "A blind reversible method for watermarking relational databases based on a time-stamping protocol," Expert Systems with Applications, vol. 39, no. 3, pp. 3185–3196, 2012.

[15] V. Khanduja, S. Chakraverty, O. P. Verma, and N. Singh, "A scheme for robust biometric watermarking in web databases for ownership proof with identification," in International Conference on Active Media Technology. Springer, 2014, pp. 212–225.

[16] D. Bhagwat, L. Chiticariu, W. C. Tan, and G. Vijayvargiya, "An annotation management system for relational databases," Vldb Journal, vol. 14, no. 4, pp. 373–396, 2005.

[17] M. Kamran and M. Farooq, "An optimized information-preserving relational database watermarking scheme for ownership protection of medical data," arXiv preprint arXiv:1801.09741, 2018.

[18] K. K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," in Conference on Usenix '06 Technical Conference, 2006, pp. 4–4.

[19] A. Prabhune, A. Zweig, R. Stotzka, J. Hesser, and M. Gertz, "P-pif: a provone provenance interoperability framework for analyzing heterogeneous workflow specifications and provenance traces," Distributed & Parallel Databases, vol. 36, no. 6, pp. 1–46, 2017.

[20] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "Scientific workflow scheduling with provenance data in a multisite cloud," in Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIII. Springer, 2017, pp. 80–112.

[21] F. Hondo, P. Wercelens, W. D. Silva, K. Castro, I. Santana, M. E. Walter, A. Araujo, M. Holanda, and S. Lifschitz, "Data provenance management for bioinformatics workflows using nosql database systems in a cloud computing environment," in IEEE International Conference on Bioinformatics and Biomedicine, 2017, pp. 1929–1934.

[22] S. B. Davidson and J. Freire, "Provenance and scientific workflows: challenges and opportunities," in Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008, pp. 1345–1350.

[23] K. K. Muniswamy-Reddy, U. Braun, D. A. Holland, P. Macko, D. Maclean, D. Margo, M. Seltzer, and R. Smogor, "Layering in provenance systems," in Conference on Usenix Technical Conference, 2009, pp. 10–10.

[24] L. Chiticariu, W. C. Tan, and G. Vijayvargiya, "Dbnotes:a post-it system for relational databases based on provenance," in ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, Usa, June, 2005, pp. 942–944.

[25] H. Fan, "Tracing data lineage using automed schema transformation pathways," in British National Conference on Databases. Springer, 2002, pp. 50–53.

[26] A. Alawini, S. Davidson, G. Silvello, V. Tannen, and Y. Wu, "Data citation: A new provenance challenge," Data Engineering, p. 27, 2018.

[27] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting relational databases: schemes and specialties," IEEE Transactions on Dependable & Secure Computing, vol. 2, no. 1, pp. 34–45, 2005.

[28] M. Zhou, J. Wang, C. Wang, and D. Li, "A novel fingerprinting architecture for relational data," in Inaugural Ieee-Ies Digital Ecosystems and Technologies Conference, 2007, pp. 477–480.

[29] J. Lafaye, D. Gross-Amblard, C. Constantin, and M. Guerrouani, "Watermill: An optimized fingerprinting system for databases under constraints," IEEE Transactions on Knowledge & Data Engineering, vol. 20, no. 4, pp. 532–546, 2008.

[30] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in Ieee/acm International Symposium on Cluster, Cloud and Grid Computing, 2017, pp. 468–477.

[31] J. L. C. Sanchez, J. B. Bernabe, and A. F. Skarmeta, "Towards privacy preserving data provenance for the internet of things," in Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on. IEEE, 2018, pp. 41–46.

[32] H. Olufowobi, R. Engel, N. Baracaldo, L. A. D. Bathen, S. Tata, and H. Ludwig, "Data provenance model for internet of things (iot) systems," in International Conference on Service-Oriented Computing. Springer, 2016, pp. 85–91.

[33] P. Senellart, "Provenance and probabilities in relational databases," ACM SIGMOD Record, vol. 46, no. 4, pp. 5–15, 2018.

[34] J. Z. Pinn and A. F. Zung, "A new watermarking technique for secure database," arXiv preprint arXiv:1304.7094, 2013.

[35] R. Halder, S. Pal, and A. Cortesi, "Watermarking techniques for relational databases: Survey, classification and comparison." J. UCS, vol. 16, no. 21, pp. 3164–3190, 2010.

[36] V. Khanduja, O. P. Verma, and S. Chakraverty, "Watermarking relational databases using bacterial foraging algorithm," Multimedia Tools and Applications, vol. 74, no. 3, pp. 813–839, 2015.

[37] N. Zawawi, R. El-Gohary, M. Hamdy, and M. F. Tolba, "A novel watermarking approach for data integrity and non-repudiation in rational databases," in International Conference on Advanced Machine Learning Technologies and Applications. Springer, 2012, pp. 532–542.

[38] G. Gupta and J. Pieprzyk, "Reversible and blind database watermarking using difference expansion," in Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 24.

[39] M. E. Farfoura and S. J. Horng, "A novel blind reversible method for watermarking relational databases," in International Symposium on Parallel and Distributed Processing with Applications, 2010, pp. 563–569.

[40] M. Shehab, E. Bertino, and A. Ghafoor, "Watermarking relational databases using optimization-based techniques," IEEE transactions on knowledge and data engineering, vol. 20, no. 1, pp. 116–129, 2008.

[41] K. Jawad and A. Khan, "Genetic algorithm and difference expansion based reversible watermarking for relational databases," Journal of Systems and Software, vol. 86, no. 11, pp. 2742–2753, 2013.

CHUNDONG WANG received the B.Sc. degree in computer science from Tianjin Normal University, China, in 1991, and the M.Sc. and Ph.D. degrees in computer science from Nankai University, China, in 2002 and 2007, respectively. He is currently a Professor with the Tianjin University of Technology. His current research interests include network information security, pervasive computing, mobile computing, and intelligent information processing.

LEI YANG received her B.S. degree from Nanjing University of Posts and Telecommunications in June 2016. She is currently working towards the M.S. degree in Information and Communication Engineering at Tianjin University of Technology, China. Her research interests are in the area of data privacy protection and database watermarking. Her technical specialties are C/C++, MATLAB and algorithms.

YIJIE WU received her B.S. degree from NanKai University Binhai College, China, in June 2016. She is currently working towards the M.S. degree in Computer Technology at NanKai University, China. Her research interests are in the area of information security and data privacy protection. Her technical specialties are C/C++, Python, and algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2876879, IEEE Access

YUDUO WU will recieve Bachelor Degree of Information Securiy and Law from Nankai University, Tianjin, China in 2019. Her research interests include applied cryptograpy, data privay protection.

XIAOCHUN CHENG (IEEE Senior Member since 2004) Xiaochun Cheng (SMą́04) received the B.Eng. degree in computer software and the Ph.D. degree in artificial intelligence from Jilin University, China, in 1992 and 1996, respectively. He is a member of the IEEE SMC: Technical Committee on Systems Safety and Security. He is also a Committee Member of the European Systems Safety Society. He is the Secretary of the IEEE SMC, United Kingdom and Republic of Ireland.

ZHAOHUI LI graduated from Nankai University in July 2003 with a degree in Control Theory and Control Engineering. He is currently working at the School of Cyberspace Security of Nankai University. His research direction is information security.

ZHELI LIU received the B.Sc. and M.Sc. degrees in computer science and the Ph.D. degree in computer applications from Jilin University, China, in 2002, 2005, and 2009, respectively. He was a Post-Doctoral Fellow with Nankai University. He joined the College of Computer and Control Engineering, Nankai University, in 2011, where he is currently an Associate Professor. His current research interests include applied cryptography and data privacy protection.